Large-Scale Forcing with Less Communication in Finite-Difference Simulations of Stationary Isotropic Turbulence

Ryo ONISHI[a,], Yuya BABA[a], Keiko TAKAHASHI[a]

[a]*Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology, 3173-25 Showa-machi, Kanazawa-ku, Yokohama Kanagawa 236-0001, Japan*

Abstract

This study proposes a new forcing scheme suitable for massively-parallel finite-difference simulations of stationary isotropic turbulence. The proposed forcing scheme, named reduced-communication forcing (RCF), is based on the same idea as the conventional large-scale forcing scheme, but requires much less data communication, leading to a high parallel efficiency. It has been confirmed that the RCF scheme works intrinsically in the same manner as the conventional large-scale forcing scheme. Comparisons have revealed that a fourth-order finite-difference model run in combination with the RCF scheme (FDM-RCF) is as good as a spectral model, while requiring less computational costs. For the range $80 < Re_\lambda < 540$, where Re_λ is the Taylor microscale-based Reynolds number, large computations using the FDM-RCF scheme show that the Reynolds dependences of skewness and flatness factors have similar power-laws as found in previous studies.

Keywords: homogeneous isotropic turbulence, finite-difference scheme, parallel computing, large-scale forcing

1. Introduction

Numerical simulations of isotropic turbulence continue to play an important role in studies of the fundamental phenomena associated with turbulence. Rapid developments in computational facilities enable us to investigate, for example, the intermittent nature of turbulence across a wide range

Email address: onishi.ryo@jamstec.go.jp (Ryo ONISHI)

of Reynolds numbers. However, there is still a need for numerical data at even higher Reynolds numbers. For example, there is currently a debate concerning the power-laws that govern the Reynolds dependence of the flatness factor of the longitudinal velocity gradient $F = \langle(\partial u/\partial x)^4\rangle \,/\, \langle(\partial u/\partial x)^2\rangle^2$, which is a measure of intermittency. Tabeling and Willaime [1] suggest there may be a transition in the power-law at around $Re_\lambda \sim 700$, where Re_λ is the Taylor microscale-based Reynolds number defined as $Re_\lambda = u' l_\lambda / \nu$, where u' is the RMS of the longitudinal velocity, l_λ is the Taylor microscale and ν is the kinematic viscosity. Specifically, they suggest that the flatness factor increases up to $Re_\lambda = 700$, but then decreases before eventually increasing again. However, the wind-tunnel experiments of Gylfason et al. [2] show no such transition. Direct numerical simulation (DNS) of turbulent flows has the potential to settle this argument because it is free from experimental ambiguities such as the effects of using Taylor's hypothesis, one-dimensional surrogates and so on. Using a cubic grid with 4096 points along each dimension, Ishihara et al. [3] achieved $Re_\lambda = 675$ with $k_{max} l_\eta \sim 2$, where k_{max} is a maximum effective wavenumber and l_η is the Kolmogorov length. Unfortunately, the Reynolds number for this large-size computation was still not sufficient to settle the argument. Ideally, we would like to increase Re_λ by a factor of 2, which demands approximately $(2^{3/2})^4 = 2^6$ times more computation. (The power of 4 comes from the sum of the three spatial dimensions and the one time dimension.) Massively parallel computing is a must for such a large computation.

Numerical simulations of isotropic turbulence are most often formulated in spectral space, since the periodic boundary conditions are then easily implemented. Spectral models are, however, facing a technical problem in massively-parallel computing. Although parallel versions of the 3D FFT algorithm are available in several scientific libraries, heavy data communication makes it difficult for spectral models to achieve a high parallel efficiency in highly parallel distributed-memory calculations. (Generally, these use the Message Passing Interface (MPI) library.) Kaneda et al. [4] carried out their spectral model simulations on the Earth Simulator - a vector-type supercomputer with a wide network bandwidth - and were able to use a 4096^3 grid; still the largest in the world 7 years later. These days, however, supercomputer systems are mostly based on scalar-type architectures with relatively narrow memory bandwidths, making it more difficult to achieve high parallel efficiency for spectral models. For this reason, finite-difference models (FDMs) have become an attractive alternative to spectral models.

Because of their inferior accuracy, FDMs have generally been less favored than spectral models. For example, Herring et al. [5] reported that FDMs require twice the resolution of spectral models to achieve the same accuracy. This is partly because many FDM simulations use upstream schemes, which contain numerical diffusion. For example, the FDM of Rai and Moin [6] uses the 5th-order upstream scheme. Past FDMs also suffered from inappropriate formulations of the finite-difference scheme, leading to non-conservation of mass, momentum and kinetic energy. Morinishi et al. [7] corrected this deficiency by formulating fully conservative high-order accurate finite-difference schemes. Kajishima et al. [8] then showed that their fourth-order central difference FDM was as good as a spectral model in simulations of decaying turbulence. In the first part of this paper we will re-examine the reliability of FDMs in comparison with spectral models.

One particular computational problem arises when the finite-difference approach is applied to stationary isotropic turbulence (which is preferable to decaying turbulence in terms of statistical analysis). The most common method for forcing the flow is to apply a forcing to the Navier-Stokes equations for all modes in the wavenumber sphere $|\mathbf{k}| < k_f$, where k_f is a cut-off wavenumber, often refered to as large-scale forcing. Michioka and Komori [9] employed this approach in their 512^3 gridpoint FDM. In order to apply the forcing, they performed Fourier transformations on the complete velocity field in order to obtain the low-wavenumber components. However, this direct application of the large-scale forcing technique reduces the parallel efficiency in larger-size computations since it requires communication of the whole velocity field.

Recently, Lundgren [10] proposed a linear forcing, where a force proportional to velocity is applied. This forcing is implemented in gridpoint rather than Fourier space, thereby avoiding the data communication associated with the Fourier transforms. Rosales and Meneveau [11] examined the properties of this scheme and reported a similar extent of the Kolmogorov -5/3 range to that achieved using the large-scale forcing method. However, they also found a smaller integral length scale, reducing the effective scaling range for a given resolution. Thus, the removal of the need for communication comes at the cost of smaller Reynolds numbers for a given resolution.

This study proposes a new forcing scheme suitable for massively-parallel FDMs. The proposed scheme is based on the same idea as the conventional large-scale forcing scheme, but requires much less data communication. We call the scheme *large-scale forcing with less communication*, or,

more simply, *reduced-communication forcing* (RCF). The RCF scheme performs volume-averaging on the velocity fields before Fourier transforming to extract the large-scale motions. Small-scale information is removed by the volume-averaging, but this is not an issue when forcing the large-scales. The size of the volume-averaged data is, of course, smaller than that of the full data, and this reduces the data communication required and leads to a high parallel efficiency. The main aims of this study are to compare the properties of the RCF scheme with those of the conventional large-scale forcing and linear forcing schemes, and to achieve high-performance computations using a fourth-order FDM with the RCF.

In the following section, we present our finite-difference model. In section 3, we describe the RCF scheme as well as the conventional large-scale forcing and linear forcing schemes. Results and discussion are presented in section 4. The FDM and spectral model are compared in subsections 4.1 and 4.2, and in subsection 4.3 the results of large-size parallel computations with our fourth-order FDM run in conjunction with the RCF are discussed. After describing the parallel efficiency of our code in subsection 4.4, the study is concluded in section 5.

2. Finite-Difference Model for the Incompressible Navier-Stokes Equations

We solve the three-dimensional continuity and Navier-Stokes equations for incompressible flows;

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_i}\left(u_i u_j\right) = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i. \tag{2}$$

Here, Re is a Reynolds number defined as $Re = U_0 L_0/\nu$, where U_0 is a representative velocity, L_0 a representative length, and ν the kinematic viscosity. We consider the case of zero mean flow, and can therefore consider the velocities in the above equations as velocity fluctuations. The last term in equation (2) represents the external forcing, which is the focus of this paper.

In our finite-difference model (FDM), spatial derivatives are calculated using fourth-order central differences, but we also ran some simulations with second-order central differences to assess the benefit of using the higher-order differences. We employed the conservative scheme of Morinishi et al. [7] for the advection term, and the second-order Runge-Kutta scheme for time integration. To solve the velocity-pressure coupling we used the HSMAC scheme [12], iterating until the RMS of the velocity divergence became smaller than δ/dx, where δ was chosen to be 10^{-3}.

The governing equations were discretized on a cubic domain of length $2\pi L_0$, and periodic boundary conditions applied in all three directions. The flow cube was discretized uniformly into N^3 gridpoints, resulting in the grid spacing $\Delta = 2\pi L_0 / N$.

3. Forcing Schemes to Obtain a Statistically Stationary State

3.1. Conventional large-scale forcing

The most common way to maintain a statistically stationary state is to force the low-wavenumber components. All Fourier modes with wavenumber $|\mathbf{k}|$ between between 0.5 and 1.5 (e.g., [13, 14]) or 0.5 and 2.5 (e.g., [15, 16, 4]) are usually forced. This restriction of the forcing to large scales allows the development of a natural inertial range at smaller scales; a white-noise random forcing may influence the intrinsic chaotic behavior of the turbulent flow.

At each timestep, the equations are first integrated without forcing, with the resulting Fourier modes denoted by $u_i^*(\mathbf{k})(t+\Delta t)$. Those Fourier modes with wavenumber $|\mathbf{k}|$ between $n-0.5$ and $n+0.5$ are then multiplied by a constant:

$$u_i(\mathbf{k})(t+\Delta t) = u_i^*(\mathbf{k})(t+\Delta t)\sqrt{E_n(0)/E_n(t+\Delta t)}, \tag{3}$$

where $E_n(t+\Delta t)$ is the energy of the modes with wavenumbers between $n-0.5$ and $n+0.5$, calculated from $u_i^*(\mathbf{k})(t+\Delta t)$, and $u_i(\mathbf{k})(t+\Delta t)$ is the updated Fourier amplitude. When forcing the Fourier modes with $|\mathbf{k}| \leq 2.5$, the modes n=1 and 2 are treated similarly.

The conventional large-scale forcing scheme is formulated in Fourier space and is therefore easily implemented in spectral models. For FDMs however, which are formulated in gridpoint space, large-scale forcing requires transformations to and from Fourier space. If full Fourier transformations are

used, covering the full spectrum, this requires data-gathering from the whole computational domain. This is not an issue for single-processor calculations ([9]), but may lead to inefficiencies when a large number of processors is used.

The efficiency of the scheme can be improved substantially, however, by restricting the transformation to those Fourier modes being forced. In this case, it is no longer necessary to gather the full velocity field from each process; all that is required is the projection of that field onto the modes being forced. For example, if (20,20,20) processes are used for a $2,000^3$ grid, and the forcing is for $0.5 < |\mathbf{k}| \leq 2.5$, the number of values to be communicated for each velocity component (u, v or w) is $20^3 \times 19 \times 2$, where the factor of 19 comes from the number of modes — i.e., (1,0,0), (1,1,0), (1,1,1), (2,0,0), (2,1,0), (2,1,1) and their permutations — and the factor of 2 reflects the use of complex mode coefficients. This data size is much less than the $2,000^3$ values required for communicating the complete set of data. However, the scheme introduced in section 3.3 reduces this data size still further, to 20^3 values.

3.2. Linear forcing

In the linear forcing scheme ([10, 11]), the external forcing is proportional to the local velocity fluctuations:

$$f_i = Qu_i, \tag{4}$$

where Q is a coefficient. The kinetic energy equation $k_e\,(= (1/2)u_iu_i)$ is then written as

$$\frac{d\left\langle k_e\right\rangle}{dt} = -\epsilon + 2Q\left\langle k_e\right\rangle, \tag{5}$$

where $\langle\ \ \rangle$ represents the mean value over the domain, and $\epsilon = \left\langle (\nu\partial u_i/\partial x_j)\,(\partial u_j/\partial x_i)\right\rangle$ is the energy dissipation rate. When Q is set to $\epsilon(t)/2\left\langle k_e(t)\right\rangle$, the RHS of equation (5) becomes zero, leading to no change of kinetic energy with time. Using a constant Q imposes an eddy turnover time $T_e = \left\langle u'^2\right\rangle/\epsilon$, where u' is the RMS of the longitudinal velocity. Alternatively, if Q is set to $\epsilon_0/2\left\langle k_e(t)\right\rangle$, with ϵ_0 constant, the average dissipation rate ϵ converges towards ϵ_0 as the flow statistics become stationary.

For parallel computations on distributed-memory computers, the former requires no data communication between processes, while the latter requires

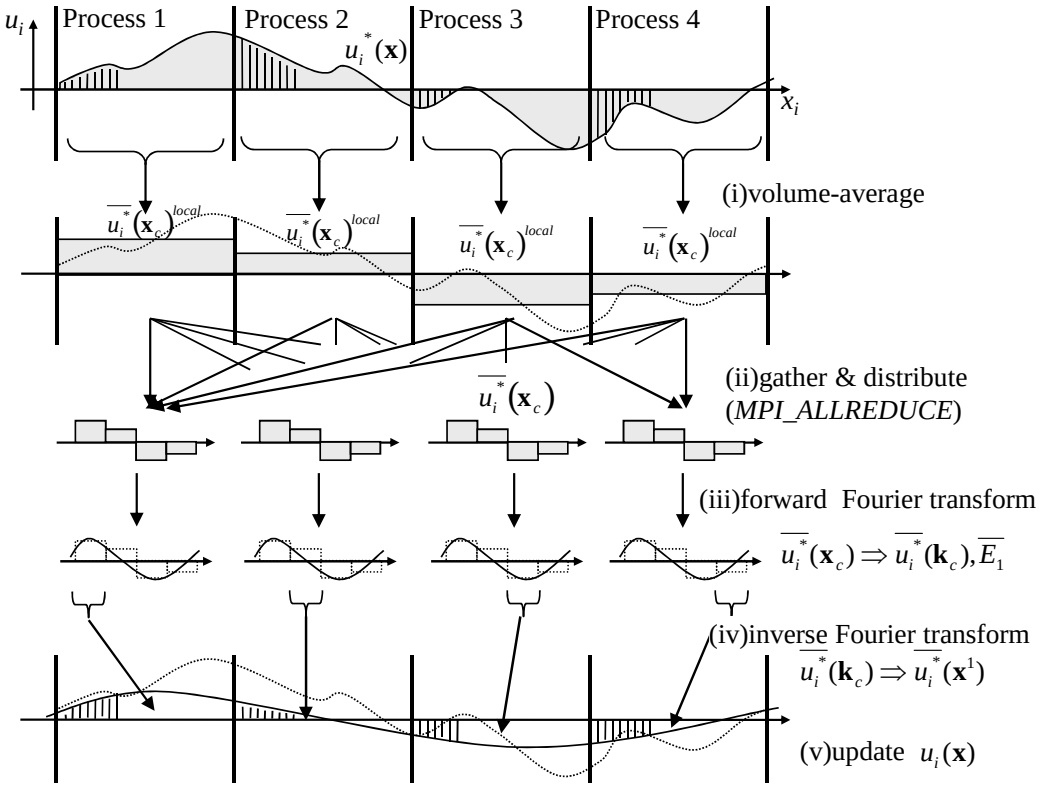


Figure 1: Schematic illustration of our reduced-communication forcing (RCF) scheme for forcing E_1. Four processes are engaged in a one-dimensional velocity calculation, so that $\mathbf{x}_c$ has dimension 4.

a small amount of communication to obtain $\langle k_e \rangle$. The locality and simplicity of the linear scheme leads to a high parallel efficiency. However, Rosales and Meneveau [11] report that the linear forcing scheme requires twice as many gridpoints to resolve the smaller scales, reducing the attainable Reynolds number.

3.3. Reduced-communication forcing (RCF)

We propose a new forcing scheme suitable for massively-parallel FDMs. The proposed scheme is based on the same idea as the conventional large-scale forcing scheme, but requires much less data communication. We call the scheme *large-scale forcing with less communication*, or, more simply, *reduced-communication forcing* (RCF). The basic idea is to volume-average the ve-

locity fields onto a coarse grid before Fourier transforming to extract the large-scale motions. The averaging process removes small-scale information, but this is not an issue when forcing the large scales. If the points that go into each volume average are all on the same processor, the data size to be communicated is then reduced to N_c^3, where N_c^3 is the number of gridpoints on the coarse grid.

The volume-averaging process can be written

$$\overline{f}(x_{c,i}^j) = \frac{1}{N/N_c} \sum_{k=1}^{N/Nc} f\left(x_i^{k+(j-1)N/N_c}\right) \quad (j = 1, ..., N_c) \tag{6}$$

where $x_{c,i}^j$ is the coordinate of the jth point on the coarse grid in the i-direction, and x_i^k that of the kth point on the original grid. Here we only consider the case where N_c is a divisor of N. For the forcing of modes $0.5 < |\mathbf{k}| \leq 1.5$, the calculation procedure for the RCF scheme is as follows. (See figure 1 for a schematic illustration.)

1. Each process volume-averages the time-integrated local velocities obtained without forcing $u_i^*(\mathbf{x}, t+\Delta t)^{local}$, giving $\overline{u_i^*}(\mathbf{x}_c, t+\Delta t)^{local}$, where $\mathbf{x}_c$ represents the coordinates of the coarse-grid system, with size N_c^3.
2. Local volume-averaged velocities $\overline{u_i^*}(\mathbf{x}_c, t+\Delta t)^{local}$ are collected and distributed using the MPI_ALLREDUCE procedure in the MPI library. All processes then obtain the volume-averaged velocities for the whole domain: $\overline{u_i^*}(\mathbf{x}_c, t+\Delta t)$.
3. Each process calculates the Fourier components of $\overline{u_i^*}(\mathbf{k}, t+\Delta t)$ using forward Fourier transformations, and then calculates the energy content $\overline{E_1}(t+\Delta t)$ of the modes with $0.5 < |\mathbf{k}| \leq 1.5$.
4. Velocities for the modes $0.5 < |\mathbf{k}| \leq 1.5$ are transformed back into grid-point space on the original grid using inverse Fourier transformations, giving $\overline{u_i^*}(\mathbf{x}, t+\Delta t)$.
5. Each process updates the velocities in order to maintain the energy in the forced modes:

$$\begin{aligned} u_i(\mathbf{x}, t+\Delta t) &= u_i^*(\mathbf{x}, t+\Delta t) \\ &+\overline{u_i^*}(\mathbf{x}, t+\Delta t)\left(\sqrt{\overline{E_1}(0)/\overline{E_1}(t+\Delta t)} - 1\right). \end{aligned} \tag{7}$$

When forcing modes with $0.5 < |\mathbf{k}| \leq 2.5$, the calculations in steps (4), (5) and (6) are extended in the obvious way.

In order to understand the affects of the volume-averaging process on the extraction of the forced modes, we can start by considering what would happen if we ignored the volume averaging and simply took point samples from the original grid at the appropriate locations. If we were to do this, the Fourier coefficients for the represented modes would become contaminated by aliasing. A common technique to reduce aliasing effects is to filter out the small scales that would alias, prior to sampling onto the coarse grid. This is effectively what the volume averaging achieves; it acts as an anti-aliasing filter.

Mathematically, the volume-averaging process in equation (6) is equivalent to the following two steps:

1. Apply a box-filter to all points on the original grid:

$$\overline{f}(x_i^j) = \frac{1}{N/N_c} \sum_{k=0}^{N/Nc-1} f(x_i^{j+k}) \quad (j = 1, ..., N) \tag{8}$$

2. Downsample onto the coarse grid:

$$\overline{f}(x_{c,i}^j) = \overline{f}(x_i^{(j-1)N/N_c+1}) \quad (j = 1, ..., N_c) \tag{9}$$

The box-filter in step 1 has the following response function in spectral space:

$$\hat{G}_{box}(k) = \frac{\sin(kL/2)}{kL/2}.$$

(There is also a phase shift of $(N/N_c - 1)/2$ gridpoints, but this is irrelevant to the procedure.) The effects of aliasing in the downsampling of step 2 modify this basic response, but the prior application of the box filter keeps this modification relatively small.

In equation (7), if we ignore the effects of aliasing, $\overline{E_1}$ is smaller than E_1 because of the non-unit response of the box filter. For example, assuming $E(k) = Ak^{-5/3}$, $\overline{E}_1 \approx \int_{0.5}^{1.5} E\hat{G}_{box}^2 dk = 0.90E_1$ if $N_c = 5$. However, note that this reduction applies to both numerator and denominator in the square-rooted term on the RHS of equation (7), so the effect cancels.

When forcing $0.5 < |\mathbf{k}| \leq 1.5$, the dimension N_c^3 of $\overline{u_i}(\mathbf{x}^c)$ must satisfy $N_c \geq \mathrm{MAX}(n_x, n_y, n_z, 3)$, where n_x, n_y and n_z are the numbers of processes in the x-, y- and z-directions. When forcing $0.5 < |\mathbf{k}| \leq 2.5$, N_c^3 must satisfy $N_c \geq \mathrm{MAX}(n_x, n_y, n_z, 5)$.

4. Results and Discussion

4.1. Decaying isotropic turbulence

Time evolutions of a flow, which initially has only large-scale motions, have been computed for five resolutions; N=32, 48, 64, 96, and 128. The initial large-scale motions have only three Fourier-vector components; (1,0,0), (0,1,0) and (0,0,1). In order to investigate the accuracy of our FDM, the flow was calculated also with a pseudo-spectral model (PSM) with the two-thirds dealiasing method ([17, 18]).

Figure 2(a) shows the time evolution of each term in the kinetic energy budget equation

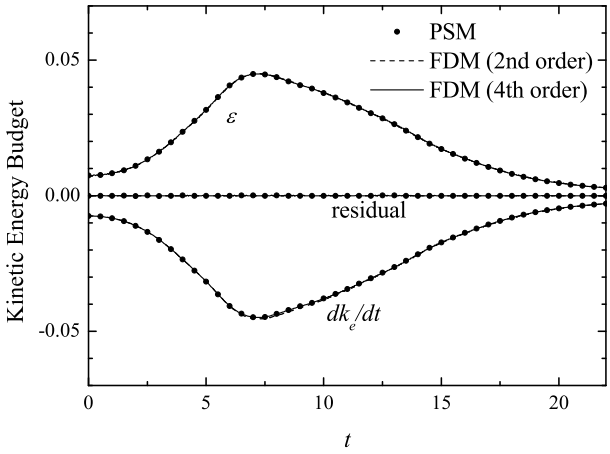
$$\frac{\partial \langle k_e \rangle}{\partial t} = -\epsilon \tag{10}$$

for the N=96 simulations. The horizontal axis shows the time normalized by L_0 and U_0, and the residual $(\partial \langle k_e \rangle / \partial t + \epsilon)$ is also shown. Results are shown for three models: the PSM, the finite-difference model with second-order differences (FDM2nd), and the finite-difference model with fourth-order differences (FDM4th). However, all the models give virtually identical results, so the lines in the plot overlap. The energy dissipation rate ϵ increases near the beginning ($t < 7$) as smaller-scale motions which can efficiently dissipate kinetic energy are generated from the larger scales through the energy cascade. The energy dissipation rate then decreases as the kinetic energy decreases.

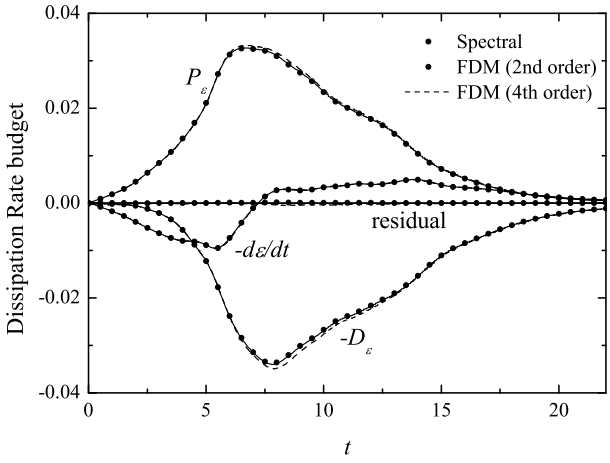
Figure 2(b) shows the time evolution of each term in the energy dissipation rate budget equation, written as

$$\frac{d\epsilon}{dt} = P_\epsilon - D_\epsilon, \tag{11}$$

where P_ϵ and D_ϵ are the production and destruction terms defined as



(a) Kinetic energy budget



(b) Energy dissipation rate budget

Figure 2: (a) Kinetic energy budget and (b) energy dissipation rate budget for three-dimensional decaying isotropic turbulence with N=96. •: pseudo-spectral model (PSM), - - - : finite-difference model with second-order central difference (FDM2nd), and —: FDM with fourth-order central differences (FDM4th).

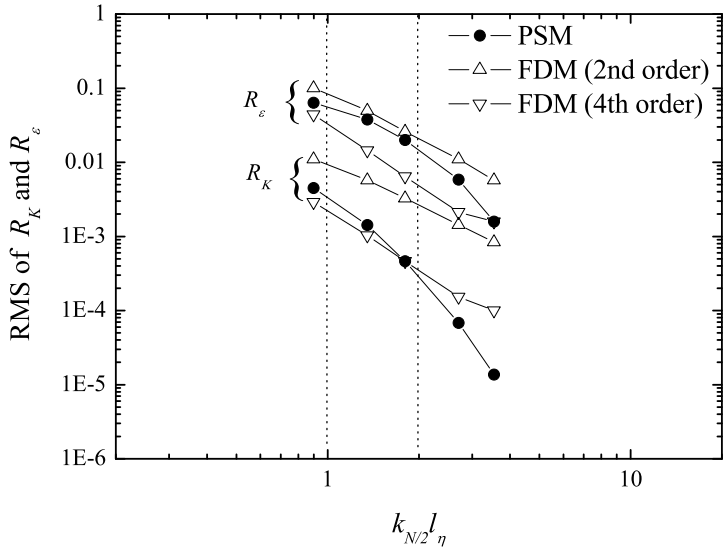


Figure 3: RMS values of normalized residuals in the kinetic energy and dissipation budgets for the five test resolutions; N=32, 48, 64, 96 and 128. -●-: PSM, -△-: FDM2nd, and -▽-: FDM4th.

$$P_\epsilon = -2\nu \left\langle \frac{\partial u_i}{\partial x_j} \frac{\partial u_i}{\partial x_k} \frac{\partial u_j}{\partial x_k} \right\rangle, \tag{12}$$

$$D_\epsilon = 2\nu^2 \left\langle \left(\frac{\partial^2 u_i}{\partial x_j \partial x_k} \right)^2 \right\rangle. \tag{13}$$

The PSM and FDM4th simulations give very similar results, but FDM2nd gives differences in P_ϵ, D_ϵ, and consequently the residual, particularly for $6 < t < 13$.

Figure 3 shows root-mean-square (RMS) values (i.e., L-2 norm) of the normalized residuals in the kinetic energy and dissipation budgets; $R_K = \left(d\left\langle k_\epsilon \right\rangle / dt + \epsilon\right) / k_e$ and $R_\epsilon = \left(d\epsilon/dt - P_\epsilon + D_\epsilon\right) / \epsilon$. The horizontal axis shows the grid resolution $k_{N/2}l_\eta$, where $k_{N/2} = N/2$ is the effective wavenumber of the FDM. Strictly speaking, the effective wavenumber of the PSM with two-thirds dealiasing is $N/3$. For comparison purposes, we adopt $k_{N/2}l_\eta$ to describe the computational resolution in this study. The RMS values of R_K are one or almost two orders of magnitude smaller than those of R_ϵ. This is

because the terms in the kinetic energy budget are second-order, while those in the energy dissipation rate budget are fourth-order. Higher-order terms induce greater numerical errors. FDM2nd shows the largest errors at all resolutions. For example, for FDM2nd, the RMS value of R_ϵ at $k_{N/2}l_\eta = 2.7$ (N=96) is almost equivalent to that of FDM4th at $k_{N/2}l_\eta = 1.4$ (N=48). This confirms that the second-order difference scheme requires twice the resolution of the fourth-order scheme to obtain comparable accuracy ([5]).

The RMS values of R_K for PSM and FDM4th are less than 10^{-2}, and negligibly small even at the coarse resolution $k_{N/2}l_\eta \sim 1$. In contrast, RMS values of R_ϵ are not negligibly small for coarse resolution; they are about 10^{-1} at $k_{N/2}l_\eta \sim 1$. A finer resolution $k_{N/2}l_\eta = 2$ is required for the energy dissipation budget calculation, which includes third- and fourth-order moments; the RMS values of R_ϵ drop to order 10^{-2} at $k_{N/2}l_\eta \sim 2$. This suggests that the coarse resolution $k_{N/2}l_\eta \sim 1$ is acceptable for low-order statistics such as the kinetic energy and energy dissipation rate, but finer resolution $k_{N/2}l_\eta \sim 2$ is required for third- or higher-order statistics ([3, 19]).

One should note that the reduction of errors is more rapid for PSM as the resolution is increased. At the coarsest resolution, $k_{N/2}l_\eta = 0.9$, the PSM has slightly larger errors than FDM4th. However, it has smaller errors for the finest resolution $k_{N/2}l_\eta = 3.5$. This can be explained by the lack of phase-errors in the spectral model; in the FDM, the phase-errors are non-zero, and become larger at small scales. Except for the very fine resolutions ($k_{N/2}l_\eta > 2$), FDM4th shows as good or better results than PSM. This result is consistent with the results of Kajishima et al. [8], who conclude that a finite-difference model with a conservative fourth-order scheme is as good as a spectral model.

4.2. Stationary isotropic turbulence

Stationary isotropic turbulence was attained using four combinations of forcing scheme and flow calculation model: (I) conventional large-scale forcing in the PSM with two-thirds dealiasing, hereafter referred to as PSM-LSF, (II) conventional large-scale forcing in the FDM; FDM-LSF, (III) reduced-communication forcing in the FDM; FDM-RCF, and (IV) linear forcing in the FDM; FDM-LF. The first three combinations are based on the same idea of forcing only the large-scale motions. In this study, the large-scale forcing injects energy in those large-scale motions with wavenumbers $|\mathbf{k}|$ between 0.5 and 1.5. In FDM-LF, the forcing coefficient Q in equation (4) was set to

$\epsilon_0/2\,\langle k_e \rangle$, where $\epsilon_0 = 0.2$. From this section onwards, all the FDMs employ conservative fourth-order centered differences for the spatial derivatives.

Figures 4(a) and 4(b) show the time evolutions of mean energy dissipation rate and Taylor-microscale based Reynolds number defined as $Re_\lambda = u' l_\lambda / \nu$, where l_λ is the Taylor microscale, for the case N=128. The horizontal axes show the nondimensional time. Both figures show that all four simulations achieve statistically stationary states at around t=15, which is about three times the eddy turnover time T_e. The FDM-RCF, PSM-LSF and FDM-LSF simulations in which only the large-scale motions are forced show similar trends, while FDM-LF produces a somewhat different behaviour. One should note that for a given resolution FDM-LF reaches significantly smaller values of Re_λ than the other simulations in figure 4(b). This is a deficiency of the linear forcing scheme, consistent with the findings of Rosales and Meneveau [11].

We note that the statistical oscillations are larger for PSM-LSF than for the FDMs. One possible reason is that the move between spectral and grid point representations causes a difference at the forcing scales, which then influences the dissipation scales. In the low Reynolds number flow considered here there is only a small separation between the forcing and dissipation scales, so this is a plausible mechanism.

Figure 4(c) shows the time evolutions of mean total energy, E_1 and $\overline{E_1}$ in FDM-RCF. As described in section 3.3, our RCF scheme ensures constant energy content in the forced wavenumber band for the volume-averaged velocities; i.e., $\overline{E_1}$ in this figure. For the raw velocities, the energy content in the forced wavenumber band, E_1, varies with time in the statistically stationary state ($t \geq 15$), with a standard deviation approximately 0.5 % of its mean value.

Figure 5 shows the stationary-state energy spectra. Each spectrum was obtained by averaging 5 sample spectra from the statistically stationary state. The spectra for PSM-LSF, FDM-LSF and FDM-RCF are almost identical, showing the Kolmogorov -5/3 power-law. FDM-LF shows similar results at high wavenumbers, but lower power at low wavenumbers. This confirms that the linear forcing has only a minor influence on the small scales despite forcing them directly to some extent. However it does not efficiently sustain the large-scale motions.

Table 1 summarizes the turbulence statistics and computational requirements for the stationary state. The turbulence statistics are written in the format [average]$\pm$[standard deviation]. As discussed above, the statistical

deviations in PSM-LSF are larger than those in the FDMs. The average dissipation rate in PSM-LSF is about 6% smaller than that in FDM-LSF and FDM-RCF, but the difference is within the standard deviations. The required numbers of floating-point operations are normalized by the values for PSM-LSF. Comparison with PSM-LSF reveals that the FDM-RCF scheme requires 35% fewer calculations. It is also shown that the FDM-RCF scheme requires slightly fewer calculations than FDM-LSF. The figures shown are for single-processor calculations. In multi-processor calculations, the difference becomes more significant due to the differing amounts of data communication. For example, we can estimate the data communication required for a calculation on a 2000^3 grid using 20^3 processors, so that each process is assigned 100^3 gridpoints. The number of values to be communicated for the spatial derivatives of each velocity component (u, v and w) is $100^2 \times 6 \times 4 = 240,000$, where 6 is the number of sides and 4 the stencil size for fourth-order central differences. The LSF scheme requires the communication of $20^3 \times 7 \times 2 = 112,000$ values for each velocity component when forcing $0.5 < |\mathbf{k}| \leq 1.5$, and $20^3 \times 19 \times 2 = 304,000$ when forcing $0.5 < |\mathbf{k}| \leq 2.5$. (See also the last paragraph in subsection 3.1). These values are comparable to that for the spatial derivatives; i.e., 240,000. In contrast, the RCF scheme requires only $20^3 = 8,000$ values in both cases, which is negligibly small compared to 240,000. We see that the LSF scheme leads to a large increase in the amount of data communication required by the FDM, while the RCF scheme does not.

The average number of iterations for each time step (including both first and second Runge-Kutta steps) was almost the same across the FDMs; 8.6 in all three variants. However, despite the similar number of floating-point operations, the percentage of computation time devoted to the HSMAC scheme was different: about 60% of elapsed time in FDM-RCF and FDM-LF, and about 40% in FDM-LSF. The difference comes from the large time used for the FFT. If we had tuned the FFT code more intensely, the HSMAC percentage for FDM-LSF would also have become around 60%. This indicates that the elapsed time depends strongly on the optimisation level of the code, and is therefore not a fair measure of the computational costs in these experiments. Thus, in Table 1 we have chosen to concentrate on floating-point operation requirements.

The results from this subsection confirm that using the RCF scheme in conjunction with the FDM works intrinsically in the same manner as the conventional large-scale forcing scheme, succeeding in producing typical sta-

	Re	ϵ	Re_λ	floating-point operations
PSM-LSF	133	0.187±0.058	69.3±6.9	1.00
FDM-LSF	133	0.199±0.025	68.3±2.6	0.668
FDM-LF	133	0.199±0.011	38.7±2.2	0.575
FDM-RCF	133	0.204±0.021	69.5±2.4	0.652

Table 1: Turbulence statistics for the stationary state (N=128), together with the computer requirements. The required numbers of floating-point operations are normalized by the values for PSM-LSF.

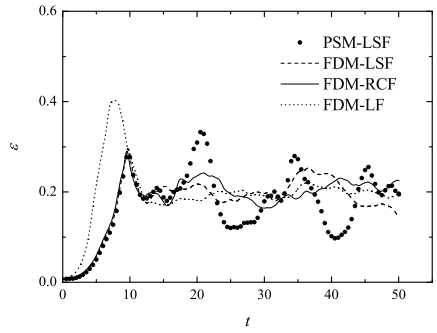
N	Re	u'	Re_λ	$k_{N/2}l_\eta$	$-S$	F
128	143	1.04	82.7	1.95	0.497	4.76
256	360	1.05	132	1.97	0.516	5.66
512	908	1.02	209	2.02	0.539	6.31
1000	2220	1.02	340	2.07	0.566	7.59
2000	5590	1.06	537	1.92	0.589	8.96

Table 2: Turbulence statistics in the stationary state from our fourth-order finite-difference model with reduced-communication forcing (FDM-RCF).

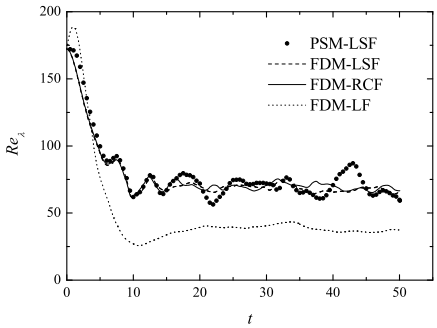
tionary isotropic turbulence. This is achieved at a smaller computational cost than the conventional combination of LSF with the spectral model.

4.3. Parallel computing for higher Reynolds numbers

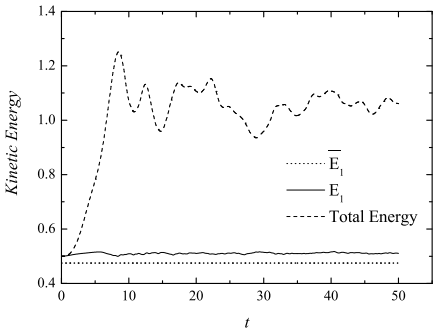
For further validation in high Reynolds number flows, the FDM-RCF calculations were extended up to $N = 2000$. Table 2 shows mean turbulence statistics obtained from the FDM-RCF model at high resolution; $k_{N/2}l_\eta \sim 2$. The statistics were averaged over more than twice the large-eddy turnover time T_e. The data fit quite well with the scalings $N \approx C_0 Re^{3/4}$ and $Re_\lambda \approx C_1 Re^{1/2}$, derived from a simple dimensional analysis based on Kolmogorov theory. The scaling constants are $C_0 = 3.1$ and $C_1 = 7.2$ in the present DNS. As shown in Figure 5, the energy spectrum for the case N=2000 has a clear Kolmogorov $k^{-5/3}$ power-law, indicating that an ideal isotropic turbulence has been formed. Table 2 also shows the skewness and flatness factors S and F of the longitudinal velocity derivatives defined by



(a) Mean energy dissipation rate



(b) Taylor-microscale-based Reynolds number



(c) Energy contents in FDM-RCF

Figure 4: Temporal evolutions of (a) mean energy dissipation rate ϵ, (b) Taylor-microscale based Reynolds number Re_λ, and (c) energy contents in FDM-RCF for the case N=128.

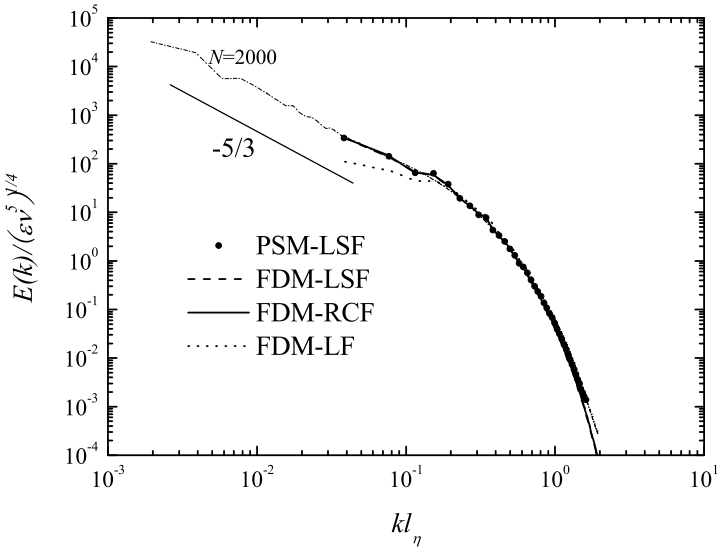


Figure 5: Energy spectra for the stationary state for N=128. •: pseudo-spectral model with large-scale forcing (PSM-LSF), - - -: finite-difference model with large-scale forcing (FDM-LSF), and —: FDM with reduced-communication forcing (FDM-RCF). The spectrum obtained from FDM-RCF with N=2000 (see subsection 4.3) is also drawn for reference (dashed-dotted line).

$$-S = \frac{\left\langle (\partial u_1/\partial x_1)^3 \right\rangle}{\left\langle (\partial u_1/\partial x_1)^2 \right\rangle^{3/2}}, \tag{14}$$

$$F = \frac{\left\langle (\partial u_1/\partial x_1)^4 \right\rangle}{\left\langle (\partial u_1/\partial x_1)^2 \right\rangle^{2}}. \tag{15}$$

On the basis of the data for $20 < Re_\lambda < 400$ from Sreenivasan and Antonia [20], and $Re_\lambda > 400$ from Antonia et al. [21], Hill [22] proposed the following power laws

$$-S \sim \begin{cases} 0.50 & (20 < Re_\lambda < 400) \\ 0.26 Re_\lambda^{0.11} & (400 < Re_\lambda), \end{cases} \tag{16}$$

$$F \sim 1.30 Re_\lambda^{0.32} \qquad (400 < Re_\lambda). \tag{17}$$

Gylfason et al. [2] then showed that

$$-S \sim 0.33 Re_\lambda^{0.09}, \tag{18}$$

$$F \sim 0.91 Re_\lambda^{0.39}. \tag{19}$$

gave a good fit to their wind-tunnel data for $100 < Re_\lambda < 900$. Recent DNSs have achieved Re_λ values of several hundred. The DNS of Ishihara et al. [3] for $100 < Re_\lambda < 700$ suggests

$$-S \sim (0.32 \pm 0.02) Re_\lambda^{0.11 \pm 0.01}, \tag{20}$$

$$F \sim (1.14 \pm 0.19) Re_\lambda^{0.34 \pm 0.03}. \tag{21}$$

Recently, Tabeling and Willaime [1] suggested that there may be a transition of the power-law for F at around $Re_\lambda \sim 700$, suggesting that the flatness factor first increases up to $Re_\lambda = 700$, but then decreases before eventually starting to increase again. They postulate that this transition is due to worm vortex breakdown, and suggest that it may be a universal characteristic of turbulence. If this is indeed the case, their suggestion poses a profound challenge to turbulence theory. Although the wind-tunnel experiments of Gylfason et al. [2] showed no evidence of the transition, it is worth investigating this issue further to better understand the fundamental nature of turbulence.

Figures 6(a) and 6(b) show the skewness and flatness factors $-S$ and F, together with results from earlier studies. The power-law increases of our skewness and flatness factors are

$$-S \sim 0.33 Re_\lambda^{0.09}, \tag{22}$$

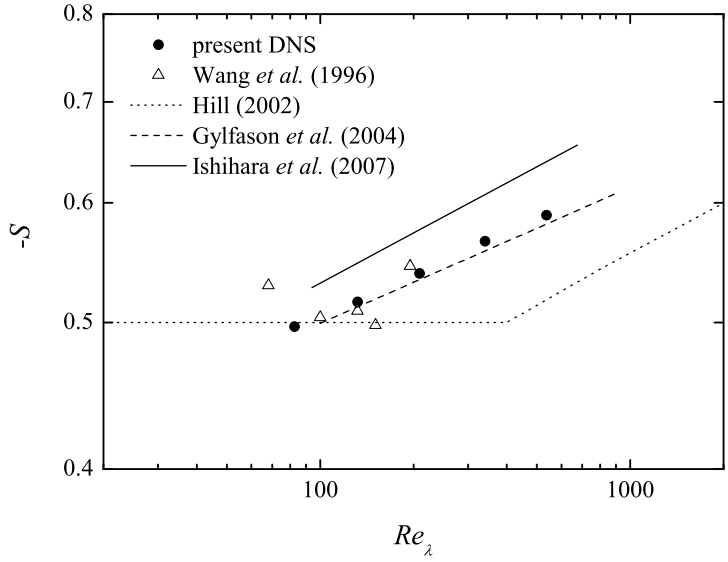
$$F \sim 1.10 Re_\lambda^{0.33}, \tag{23}$$

which are consistent with past studies in the range $80 < Re_\lambda < 540$. Unfortunately, the calculated range of Re_λ is not large enough to settle the debate on the power-law transition of the flatness factor at around $Re_\lambda = 700$. Even larger size computations are required to resolve this fundamental question.

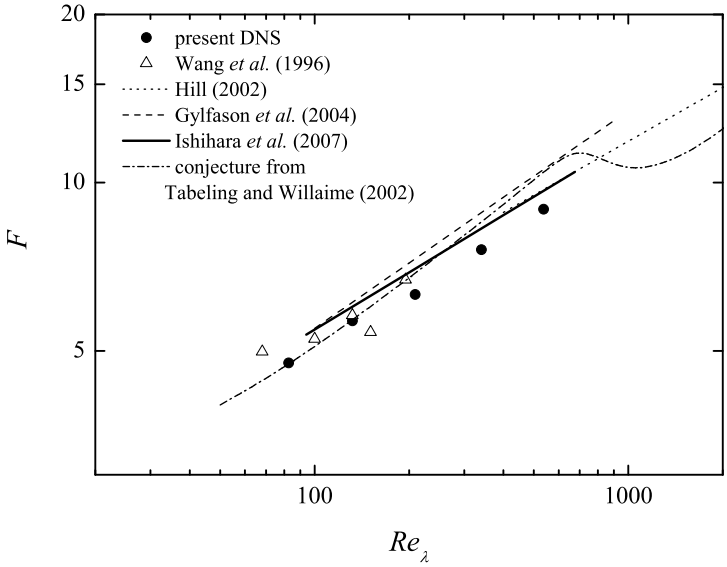
4.4. Parallel efficiency

The FDM-RCF scheme has been coded for MPI (Message Passing Interface) parallel computing. The calculation for N=2000 was run at 3.66 TFLOPS on 16 nodes (128 processors) of the Earth Simulator 2 (ES2) at the Japan Agency of Marine-Earth Science and Technology, which is about 27.9% of the peak performance of the nodes. For this value of N, each timestep took 2.3 seconds. The parallel efficiency α is defined as $\alpha = (1 - T_{N2}/T_{N1})/(1 - N_1/N_2)$, where T_{N1} and T_{N2} are the elapsed times for simulations using N_1 and N_2 ($> N_1$) processors. Comparison between the runs with 16 and 32 nodes gave $\alpha = 99.92\%$. If this parallel efficiency were maintained up to 160 nodes, we would attain a speed-up rate $(T_{N16}/T_{N160}) \times (16/160)$ of more than 0.99. This would correspond to usage of all 160 nodes of ES2, which provides a peak-performance of 131 TFLOPS. If we were to run the FDM-RCF scheme using the whole machine, a calculation for $N = 5000$ would be feasible. This would take about 3.6 seconds for each timestep, and consequently about one day to obtain the statistics. The calculation for N=5,000 with the fine resolution $k_{N/2} l_\eta \sim 2$ would provide the statistics for $Re_\lambda \sim 1,000$, which could give a hint as to whether there is indeed a transition of the flatness factor power law around $Re_\lambda = 700$.

In order to investigate the parallel efficiency in a cluster system (scalar-type supercomputer), we also ran FDM-RCF simulations on up to 256 processors of an SGI Altix4700. The mean performance attained was 7.50 % of the peak, with a mean parallel efficiency over 99.99%.



(a) Skewness



(b) Flatness

Figure 6: (a) Skewness S and (b) flatness F of the longitudinal velocity gradient $\partial u/\partial x$ for different Reynolds numbers Re_λ.

5. Conclusions

This study proposes a new simple forcing scheme suitable for massively-parallel finite-difference simulations of stationary isotropic turbulence. The proposed forcing scheme, named reduced-communication forcing (RCF), is based on the same idea as the conventional large-scale forcing (LSF) scheme, but requires much less data communication. The RCF performs volume-averaging on the velocity fields before applying Fourier transforms to extract the large-scale motions. Small-scale information is removed by volume-averaging the data, but this is not an issue when forcing the large scales. The size of the volume-averaged data is of course smaller than that of the the full set of data values, which reduces the data communications required for parallel computing and leads to a high parallel efficiency.

It is confirmed that a finite-difference model (FDM) adopting a conservative fourth-order scheme is as good as a conventional spectral model for $k_{N/2}l_\eta \leq 2$, and that the RCF scheme works intrinsically in the same manner as the LSF scheme. It is also confirmed that the combination of this FDM with the RCF scheme (FDM-RCF) can produce typical stationary isotropic turbulence at a smaller computational cost than the conventional combination of a spectral model with the LSF scheme. We therefore conclude that the FDM-RCF scheme is a promising tool for massively-parallel simulations of stationary isotropic turbulence, which will contribute to a better understanding of small-scale structures such as intermittency.

For example, a calculation for N=5,000 with fine resolution ($k_{N/2}l_\eta \sim 2$) is feasible if run using the FDM-RCF scheme on the whole of Earth Simulator 2 in the Japan Agency for Marine-Earth Science and Technology. This simulation would provide the statistics for $Re_\lambda \sim 1,000$, which would help settle the question of whether there is a transition in the power-law for the flatness factor at around $Re_\lambda = 700$.

Acknowledgments

This research was supported by a Grant-in-Aid from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan for Young Scientists (B) 21760143, and was partially supported by the Core Research for Evolutional Science and Technology (CREST) Program "Advanced Model Development and Simulations for Disaster Countermeasures" of the Japan Science and Technology Agency (JST). The large-size numerical simulations

presented were carried out on the Earth Simulator 2 supercomputer in the Japan Agency for Marine-Earth Science and Technology.

References

[1] P. Tabeling, H. Willaime, Physical Review E 65 (2002) 66301.

[2] A. Gylfason, S. Ayyalasomayajula, Z. Warhaft, Journal of Fluid Mechanics 501 (2004) 213–229.

[3] T. Ishihara, Y. Kaneda, M. Yokokawa, K. Itakura, A. Uno, Journal of Fluid Mechanics 592 (2007) 335–366.

[4] Y. Kaneda, T. Ishihara, M. Yokokawa, A. Uno, Physics of Fluids 15 (2003) L21.

[5] J. Herring, S. Orszag, R. Kraichnan, D. Fox, Journal of Fluid Mechanics 66 (1974) 417–444.

[6] M. M. Rai, P. Moin, Journal of computational physics 96 (1991) 15–53.

[7] Y. Morinishi, T. Lund, O. Vasilyev, P. Moin, Journal of Computational Physics 143 (1998) 90–124.

[8] T. Kajishima, T. Ohta, K. Okazaki, Y. Miyake, JSME International Journal 41B (1998) 830–839.

[9] T. Michioka, S. Komori, AIChE journal 50 (2004) 2705–2720.

[10] T. Lundgren, Center for Turbulence Research Annual Reserch Briefs (2003) 461–473.

[11] C. Rosales, C. Meneveau, Physics of Fluids 17 (2005) 095106.

[12] C. W. Hirt, J. L. Cook, J. Comp. Phys. 10 (1972) 324–340.

[13] R. Kerr, Phys. Fluids Suppl 59 (1985) 6735–6739.

[14] A. Vincent, M. Meneguzzi, Journal of Fluid Mechanics 225 (1991) 1–20.

[15] S. Chen, G. D. Doolen, R. H. Kraichnan, Z.-S. She, Physics of Fluids A 5 (1992) 458–463.

[16] J. G. B. Lian-Ping Wang, Shiyi Chen, J. C. Wyngaard, Journal of Fluid Mechanics 309 (1996) 113–156.

[17] K. Ishioka, GFD Dennou Club 1 (2005) 1.

[18] R. Onishi, K. Takahashi, S. Komori, Physics of Fluids 21 (2009) 125108.

[19] Y. Yamazaki, T. Ishihara, Y. Kaneda, JOURNAL-PHYSICAL SOCIETY OF JAPAN 71 (2002) 777–781.

[20] K. Sreenivasan, R. Antonia, Annual review of fluid mechanics 29 (1997) 435–472.

[21] R. Antonia, A. Chambers, B. Satyaprakash, Boundary-Layer Meteorology 21 (1981) 159–171.

[22] R. Hill, Journal of Fluid Mechanics 452 (2002) 361–370.